

## **APPENDIX A**

### **WORK PACKAGES**

## **APPENDIX A**

### **WORK PACKAGES**

This appendix contains information about each work package and its Logical Unit(s) (LUs). Each logical unit can contain one or more Functional Units (FUs) and these FUs are described in Appendix B, Functional Units.

#### **A.1 RDACargo**

This work package provides the user interface and functions for editing ULN cargo details. Its functionality supports the display and editing of 1st through 4th level detail for the ULNs in the collection.

Included in this work package is the following LU:

CargoEditor - This is the only LU in this work package. All cargo editing functionality is provided here.

The work package architecture is shown in Figure A.1-1, RDACargo Architecture.

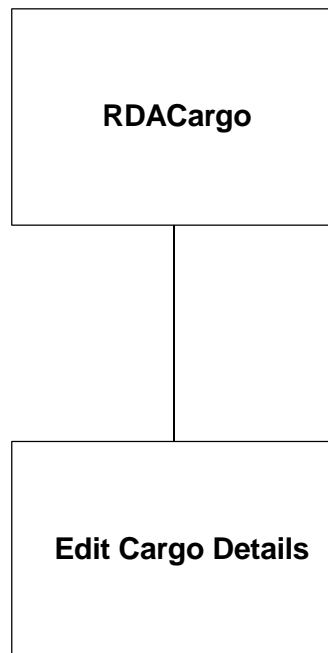


Figure A.1-1. RDACargo Architecture

This work package consists of a single logical unit which is composed of a single functional unit. The FU is CargoEditor. CargoEditor provides the user interface and functions for manipulating 1st through 4th level cargo detail.

## A.2 RDADevTools

This work package contains a number of tools developed specifically for the Gain Developer to use in debugging, analysis, and manipulation of Gain Momentum applications. The tools are intended for internal use within a development organization and not for use by the general user community.

Included in this work package are the following LUs:

- a. Database Tools - These tools provide the ability to browse database tables, execute database SQL commands, and monitor the connections opened between a Gain application and the database.
- b. Gain Tools - These tools provide the ability to browse Gain applications and objects.
- c. Miscellaneous Tools - These tools are a collection of assorted tools that do not fit comfortably into any other category.
- d. System Tools - These tools provide shortcuts and monitoring capabilities with the operating system.

The work package architecture is shown in Figure A.2-1, RDADevTools Architecture.

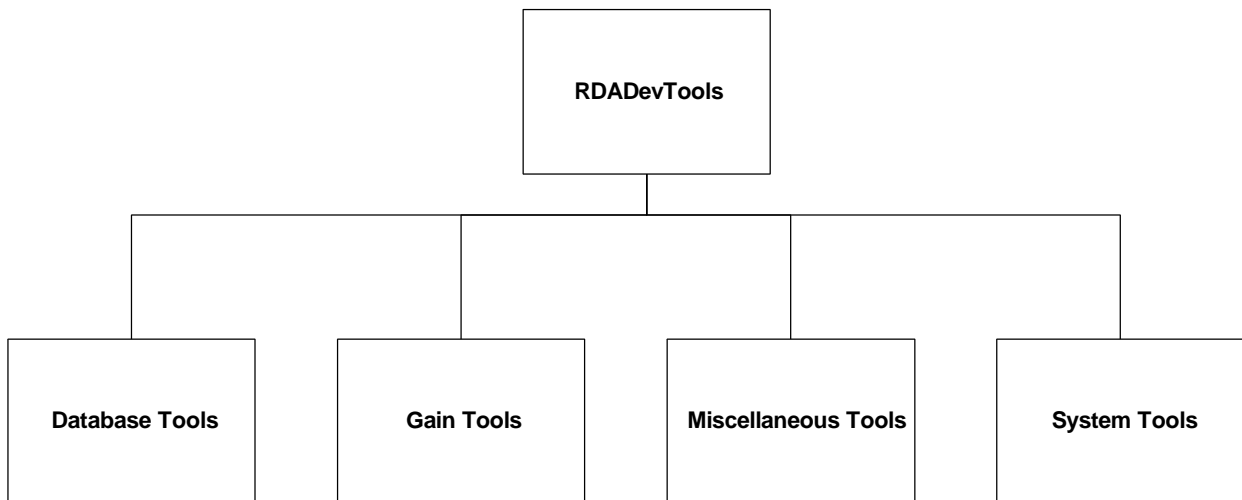


Figure A.2-1. RDADevTools Architecture

### **A.2.1 Database Tools**

This LU provides a collection of tools that can be used to browse database tables, execute ad hoc SQL commands, monitor database connections between Gain and the database, and manipulate the OPS\$ account environment variables necessary for supporting ORACLE OPS\$ accounts.

Included in this LU are the following FUs:

- a. OPS\$Tool,
- b. SQLOpsTool,
- c. SQLTester,
- d. TableViewer,
- e. TKProfTool, and
- f. XTPTTool.

### **A.2.2 Gain Tools**

This LU provides a collection of tools for working with Gain applications.

Included in this LU are the following FUs:

- a. ArrayManager,
- b. BrowserTool,
- c. EngineHelper,
- d. Finder,
- e. GainMetrics,
- f. grep,
- g. MiscOps,
- h. ObjectSelect,
- i. PagePrivTool,
- j. ProfileTool,
- k. PropertyManager,
- l. ScriptTool,
- m. SelectHelper,
- n. SystemMonitor, and
- o. ViewCheckOuts.

### **A.2.3 Miscellaneous Tools**

This LU provides a miscellaneous collection of tools that support RDA.

This LU is composed of the following FUs:

- a. ButtonBar,
- b. Colors,
- c. Pscr, and
- d. RQMTList.

#### **A.2.4 System Tools**

This LU provides a collection of tools for interacting with the operating system.

This LU is composed of the following FUs:

- a. FileSelector,
- b. Heap, and
- c. PrintMonitor.

#### **A.3 RDAForceModules**

This work package provides functions for force module editing, creation, deletion, and flow analysis. The user is able to edit the description of a force module as well as change the contents of the force module. Functions are also provided for copying force modules between TPFDDs. The flow analysis functions allow a user to store and retrieve capability limits and apply those limits to the cargo associated with one or more force modules.

Included in this work package are the following LUs:

- |                                    |   |
|------------------------------------|---|
| a. Edit Force Modules/FM Ops -     | This LU provides the user interface and functions for creating, editing, and deleting force modules. Functions are also provided for supporting the force module portions of merging TPFDDs, and copying requirements to target TPFDDs. |
| b. Provide Force Module Graphics - | This LU provides the user interface and functions for analyzing cargo flow associated with one or more force modules.   |

The work package architecture is shown in Figure A.3-1, RDAForceModules Architecture.

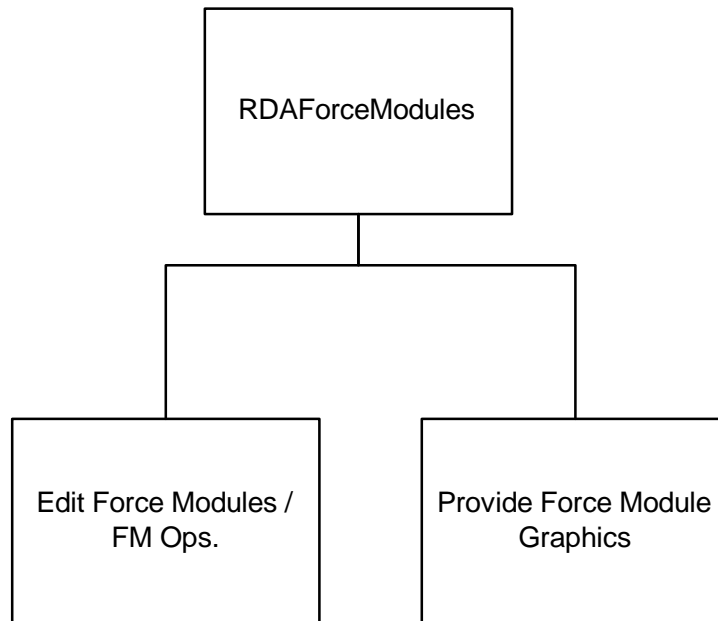


Figure A.3-1. RDAForceModules Architecture

#### **A.3.1 Edit Force Modules/FM Ops**

This LU provides all functions necessary for manipulating force modules.

Included in this LU is the following FU: EditForceModule.

#### **A.3.2 Provide Force Module Graphics**

This LU provides all the functions necessary for performing flow analysis of force module cargo. Functions are provided for creating, using, and deleting capability lines that represent predetermined flow limits. The user can analyze the flow analysis for a single force module or multiple force modules. When analyzing single force modules, the flow for each of the transportation dates can be examined.

Included in this LU are the following FUs:

- a. FMData,
- b. FMGraphics,
- c. FMlist, and
- d. FMwindows.

#### A.4 RDAMain

This work package is the main driver for the RDA System. It provides functions for viewing the available TPFDDs, launching TPFDD based and requirement based operations, and launching reference file viewing functions. TPFDD based functions are those functions that deal with TPFDDs as a whole. Requirement based operations are those operations that manipulate portions of a TPFDD.

Included in this work package are the following LUs:

- |    |   |  |
|----|---|--|
| a. | Provide IRM Interface -                 | This LU provides functions that are associated with, or have a dependency upon the Information Resource Management (IRM) portion of the System Services segment in the GCCS System. Currently, the RDA System relies on IRM to initialize new TPFDDs before they can be used within RDA. |
| b. | Provide Timeline Node Date Operations - | This LU provides functions for shifting the dates (Ready to Load (RLD), etc.) of a set of requirements. This is the only portion of RDA that is dependent upon the XTP segment of GCCS.  |
| c. | Provide TPFDD/OPLAN Summary -           | This LU provides functions for viewing summary information about an operational plan (OPLAN) and the associated TPFDD, as well as editing the narrative information associated with an OPLAN.  |
| d. | RDASkeleton -                           | This LU provides the main backbone or structure on which the rest of the RDA System exists.  |
| e. | Update PID from TUCHA -                 | This LU provides the capability to update the Unit Type Code (UTC) associated information for a selected plan ID (PID) with new UTC information found in the Type Units Characteristics (TUCHA) reference file.  |

The work package architecture is shown in Figure A.4-1, RDAMain Architecture.

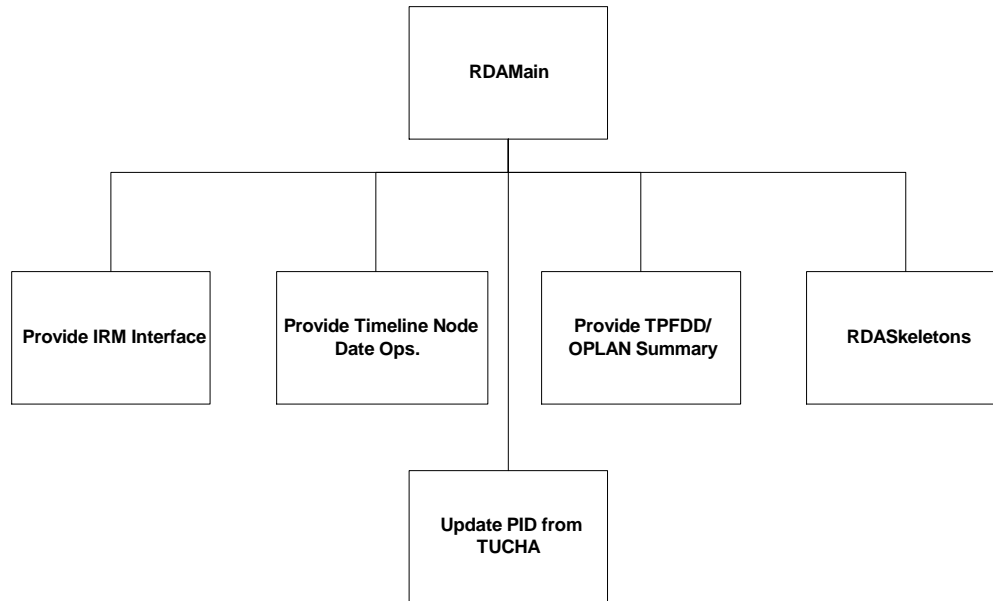


Figure A.4-1. RDAMain Architecture

#### A.4.1 Provide IRM Interface

This LU acts as the holder of links between RDA and System Services. Currently there are no explicit links between the two systems. Current functionality in the LU is to provide the ability to copy a plan into a destination plan.

This LU is composed of the following FU: CopyPlan.



#### **A.4.2 Provide Timeline Node Date Ops**

This LU provides the functions necessary for shifting TPFDD dates (RLD, etc.) by a set number of days (positive or negative). This can be done for the contents of the collection, the marked records, or for any set of force modules. Processing is accomplished through the use of the XTP.

This LU is composed of the following FU: ShiftTPFDDdates

#### **A.4.3 Provide TPFDD/OPLAN Summary**

This LU provides the functions necessary for viewing summary information about an OPLAN and editing the OPLAN narrative information.

This LU is composed of the following FU: PlanSummary.

#### **A.4.4 RDASkeletons**

This LU provides the skeleton upon which all the other RDA functions rest. Functions are provided for initializing RDA and for managing the display of the main work area for editing requirements in RDA.

This LU is composed of the following FUs:

- a. Editor, and
- b. RDASkeleton.

#### **A.4.5 Update PID from TUCHA**

This LU provides the functions necessary for updating a selected TPFDD with current TUCHA information. These updates take the form of modifications to the UTCs used by the requirements in the TPFDD. UTCs may be deleted or changed.

This LU is composed of the following FU: UpdatePIDfromTucha.

#### **A.5 RDAMapping**

This work package will be implemented in a future release.

#### **A.6 RDAMergeCompare**

This work package provides two of the TPFDD based functions (Merging two TPFDDs and Comparing two TPFDDs). The Merger function provides the ability to merge the current TPFDD into a target TPFDD. The compare function allows the user to choose the fields and records that will be compared between two TPFDDs.

Included in this work package is the following LU:

- |                            |   |
|----------------------------|---|
| Merge and Compare TPFDDs - | This is the only LU in this work package. All merge and compare functionality is provided here. |
|----------------------------|---|

The LU and its corresponding files are described in detail below.

The work package architecture is shown in Figure A.6-1, RDAMergeCompare Architecture.

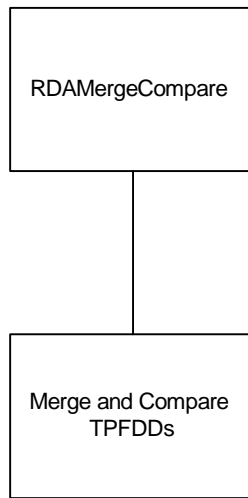


Figure A.6-1. RDAMergeCompare Architecture

This LU is composed of the following FUs:

- a. ComparePIDs, and
- b. MergePIDs.

#### **A.7 RDARefereceFiles**

This work package provides the ability to view the standard reference files in the JOPES Core database. This is a search and view only capability.

Included in this work package is the following LU:

View Reference Files -	This LU provides the user interface and functionality required to allow the user to search through the available reference files and display detailed data pertaining to records of interest. Reference file information is displayed in both a “spreadsheet” like format and a form based format. These viewers are implemented using a mixture of Gain’s datamanager functions and SQL calls.
------------------------	---

The LU and its corresponding files are described in detail below.

The work package architecture is shown in Figure A.7-1, RDAReferenceFiles Architecture.

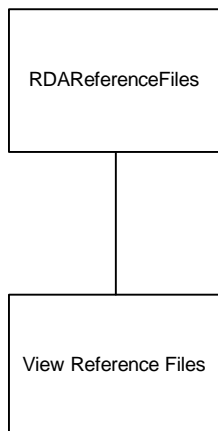


Figure A.7-1. RDAReferenceFiles Architecture

The single LU in this work package is composed of the following FUs:

- a. ViewGeoloc,
- b. ViewPorts,
- c. ViewTucha, and
- d. ViewTudet.

## A.8 RDARequirements

This work package provides the capability to create, edit, and delete requirements. Functions are provided for manipulating individual requirements, as well as performing range changes on a collection of requirements. Functions are also provided for copying requirements to other TPFDDs.

Included in this work package are the following LUs:

- |    |  |  |
|----|--|--|
| a. | Create Requirements -                  | This LU provides user interface and functions for creating requirements in a TPFDD.                                      |
| b. | Edit Entire Collection/Marked Record - | This LU provides user interface and functions for making range changes to sets of requirements.                          |
| c. | Edit Individual Requirements -         | This LU provides user interface and functions for modifying individual requirements.                                     |
| d. | Provide Renumbering -                  | This LU provides user interface and functions for changing the requirement IDs for individual and multiple requirements. |

The work package architecture is shown in Figure A.8-1, RDARRequirements Architecture.

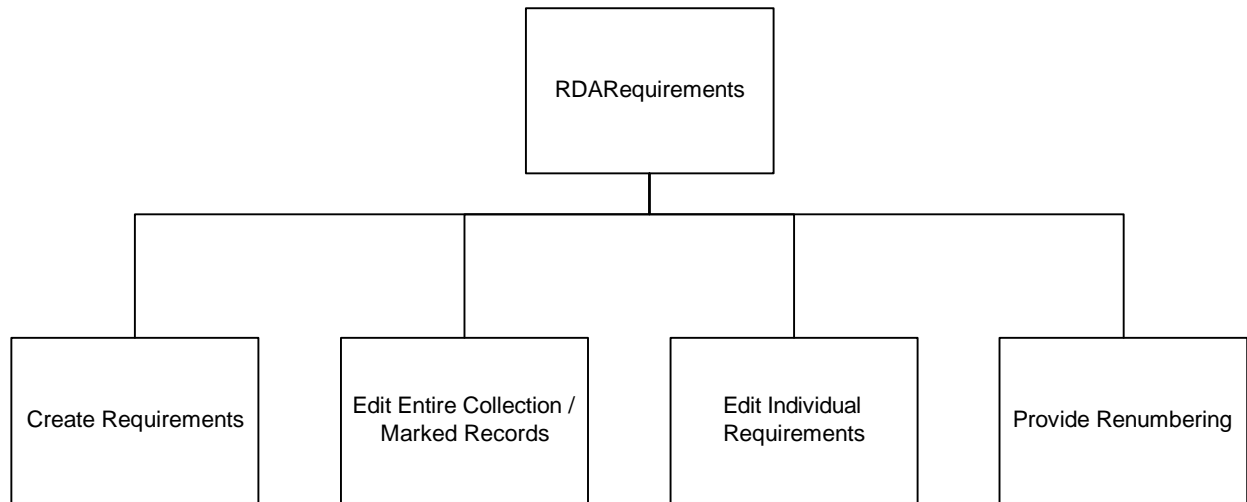


Figure A.8-1. RDARRequirements Architecture

### A.8.1 Create Requirements

This LU provides the user interface and functions for creating new ULNs, CINs, and PINs in the current TPFDD. Up to 99 new requirements can be created, and the user can choose from a number of different numbering styles for creating the requirement IDs. Support is also provided for creating parent units.

This LU is composed of the following FUs:

- a. CreateCINsPINs,
- b. CreateRecs, and
- c. CreateULNs.

### A.8.2 Edit Entire Collection/Marked Records

This LU provides the user interface and functions for performing range changes to the collection and to marked records. These changes result in mass updates to the database.

This LU is composed of the following FUs:

- a. CollectionDetail, and
- b. Marked\_Records.

### **A.8.3 Edit Individual Requirements**

This LU provides the user interface and functions for performing edits to individual requirements (ULNs, CINs, and PINs).

This LU is composed of the following FUs:

- a. CINDetails,
- b. EditRecords,
- c. PINDetails,
- d. SplitShipments,
- e. ULNDetail,
- f. ULNTransactions, and
- g. UnsplitShipments.

### **A.8.4 Provide Renumbering**

This LU provides the user interface and functions necessary for renumbering requirements (ULNs, CINs, and PINs). The design allows the user to configure the renumbering operation according to a predetermined set of basic rules for numbering each position of the 7 character ID. Preconfigured numbering schemes are provided and can be used as a starting point, or used as is.

This LU is composed of the following FU: RenumberOps.

## **A.9 RDASelect**

This work package provides the capability to populate the RDA collection with the results of a query that is executed against the core database. No knowledge of the database structure is required, and the user can store and retrieve queries for later use. The queries are stored in a location in the core database that is shared with the Ad Hoc Query application.

Included in this work package is the following LU:

- |                                    |  |
|------------------------------------|--|
| Provide User Selected Retrievals - | This LU provides the user interface and functions necessary for constructing the database queries. |
|------------------------------------|--|

The work package architecture is shown in Figure A.9-1, RDASelect Architecture.

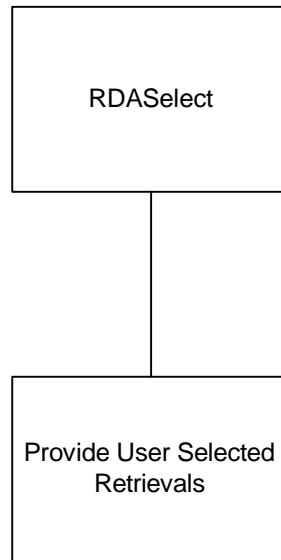


Figure A.9-1. RDASelect Architecture

The LU is composed of a single FU, Select. Select provides all the functions necessary for retrieving and storing queries, generating queries and populating the RDA collection table with the results of the queries. Queries are generated as a series of 20 qualification statements and can include any combination of ULNs, CINs, and PINs. Support is provided for up to 4 sort fields in ascending or descending order. When applying a query result to an existing collection, the user is able to replace the existing collection or add the result to the existing collection. When adding the result, the collection is resorted according to the sort criteria for the last sort. Sort keys are generated by copying sort key data to the collection table.

#### **A.10 RDATimeline**

This work package provides the capability to graphically view and manipulate schedules for the requirements that are in the current collection. Functions are provided for viewing, zooming, editing scheduling details, marking, and viewing miniature time lines.

Included in this work package is the following LU:

Timeline Node and Link Operations -	This LU contains all the functionality required for this work package.
-------------------------------------	--

The work package architecture is shown in Figure A.10-1, RDATimeline Architecture.

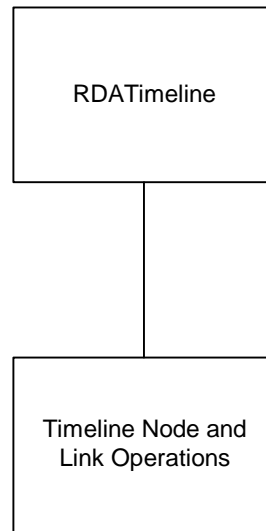


Figure A.10-1. RDATimeline Architecture

This LU is composed of the following FUs:

- a. RDATimeline,
- b. TimelineWindows, and
- c. TLUlnList.

### A.11 RDAUtility

This work package provides a set of commonly used functions that are shared by one or more other Functional Units in the system. This work package is intended to be a reuse library that can be utilized by any other Gain application.

Included in this work package are the following LUs:

- a. Database Utilities - This LU provides a set of reusable functions and utilities that can be used in Gain applications that access an ORACLE database.
- b. Look Up Utilities - This LU provides a set of functions for displaying a pick list of choices to the user and allowing them to select one or more entries. Functions are also provided for performing limited searches of selected reference files used to fill fields in the RDA System.
- c. System Utilities - This LU provides a set of functions for making it easier to interact with the operating system and for standardizing the display and processing of ORACLE and Gain errors.

- d. Verification Engine - This LU provides a function for executing the verification engine stored procedures.

The work package architecture is shown in Figure A.11-1, RDAUtility Architecture.

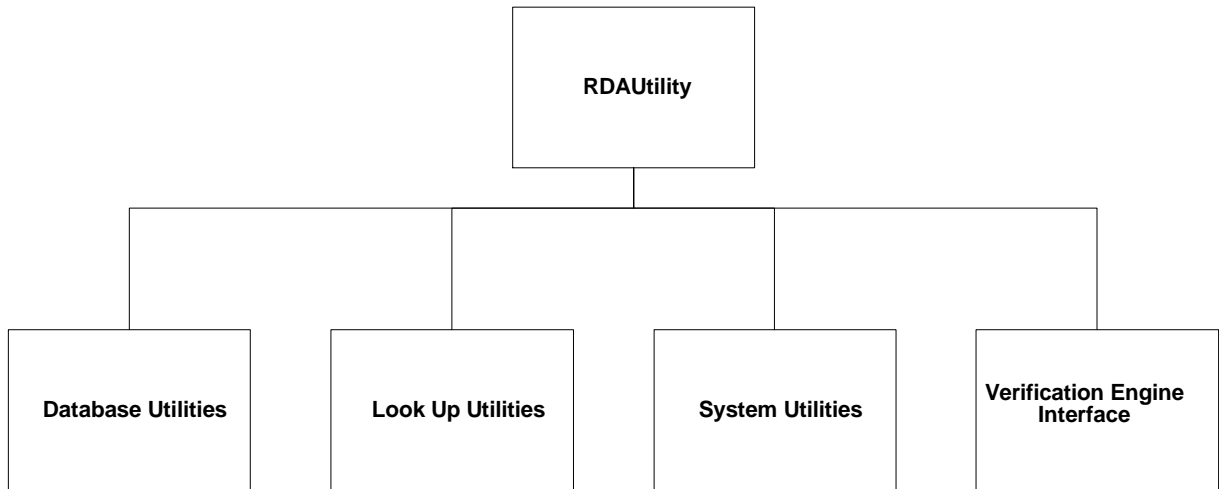


Figure A.11-1. RDAUtility Architecture

#### A.11.1 Database Utilities

This LU provides a collection of functions for creating and controlling connections to an ORACLE database. These utilities are different from those found in RDADevTools in that these are intended to be production code that are used in applications.

This LU is composed of the following FUs:

- a. RDACollectionCheck,
- b. SQLAccess, and
- c. SQLError.

#### A.11.2 Look Up Utilities

This LU provides a collection of utilities that are used throughout the RDA System to provide the user with pick lists and search capabilities for filling in requirement data fields. Utilities range from being able to pick an entry from a list to being able to search a reference file for qualifying records and pick one or more entries.



This LU is composed of the following FUs:

- a. CargoSelect,
- b. EquipmentLookup,
- c. ErrorSelect,
- d. GEOSelect,
- e. GSORTSSelect,
- f. LookUpApp,
- g. TargetPIDSelect, and
- h. TUCHASelect.

### **A.11.3 System Utilities**

This LU contains utilities that are intended for general use by all portions of the RDA System. Utilities in this category are used by the majority of the RDA System.

This LU is composed of the following FUs:

- a. ProgressBar,
- b. RDAError,
- c. RDAToolBox, and
- d. TestApp.

### **A.11.4 Verification Engine Interface**

This LU contains a single FU which is the Gain interface for calling the Verification Engine stored procedures on individual as well as all requirements in a given plan.

This LU is composed of the following FU: Verification Engine Interface.

## **A.12 Transaction Generation**

This work package provides ORACLE stored procedures for generating JDS formatted transactions. These transactions are deposited in the send queue for distribution to other sites. Transaction generation is used by any function that creates, updates, or deletes requirements related data.

## **A.13 Verification Engine**

This work package provides a central set of ORACLE stored procedures that will perform logical error checks against data in the database. Checks are provided for TPFDD information, Force Module information, and requirement information (Unit Line Numbers (ULNs), Cargo Line Numbers (CINs), and Personnel Line Numbers (PINs)). The contents of this work package are accessible to any system that can access the database.

Included in this work package is the following LU:

Verification Engine - This LU contains the stored procedures for performing all the logical error checks in the database.

The work package architecture is shown in Figure A.13-1, Verification Engine Architecture.

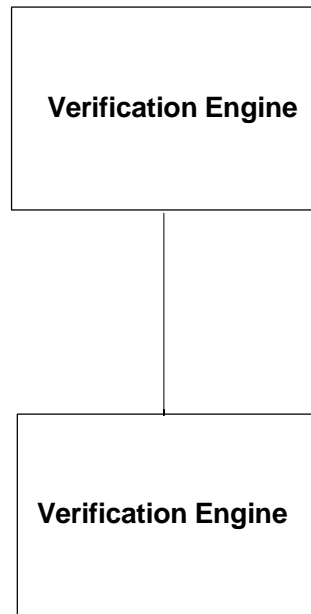


Figure A.13-1. Verification Engine Architecture

The single LU is composed of the following FUs:

- a. rda\_pk\_ve1,
- b. rda\_pk\_ve2,
- c. rda\_pk\_ve3,
- d. rda\_pk\_ve4,
- e. rda\_pk\_ve5,
- f. rda\_pk\_ve6, and
- g. rda\_pk\_ve7.